CHAPTER 6

Service Design

From the Home window of Cisco IP Solution Center (ISC), which you receive upon logging in, click the **Service Design** tab and you receive a window as shown in Figure 6-1, "Service Design Selections."

Figure 6-1 Service Design Selections

Next you can choose the following selections:

- **Policies, page 6-1** Create and manage Policies for licensed services.
- **Templates, page 6-2** Create and manage Templates and associated data.

Policies

Policies is explained in each of the *User Guides* for each of the applicable licensed services.

Templates

Templates supports the browsing, creation, and deletion of Template Folders, Templates, and Data Files and it supports the viewing of Template-generated configurations. The configuration created from the template and data file can be downloaded to devices. When creating a Service Request, you can select from the list of templates and data files and associate them with the Service Request. At Deploy time, the template and data file are instantiated and the configuration is appended or prepended to the configlet generated by ISC. Another method is to use the Device Console feature to download templates independent of Service Requests, as explained in the "Download Template" section on page 5-3.

ISC provides a way to integrate a template with ISC configlets.

For a given customer edge router and/or provider edge router, you specify the following:

- template name
- template data file name
- whether the template configuration file should be appended or prepended to the ISC configlet
- whether the template configuration file is active or inactive for downloading to the edge device

The template data files are tightly linked with the corresponding template. You can use a data file and its associated template to create a template configuration file. The template configuration file is merged with (either appended or prepended to) the ISC configlet. ISC downloads the combined ISC configlet and template configuration file to the edge device router.

- You can download a template configuration file to a router.
- You can apply the same template to multiple edge routers, assigning the appropriate template data file for each device. Each template data file includes the specific data for a particular device (for example, the management IP address or host name of each device).

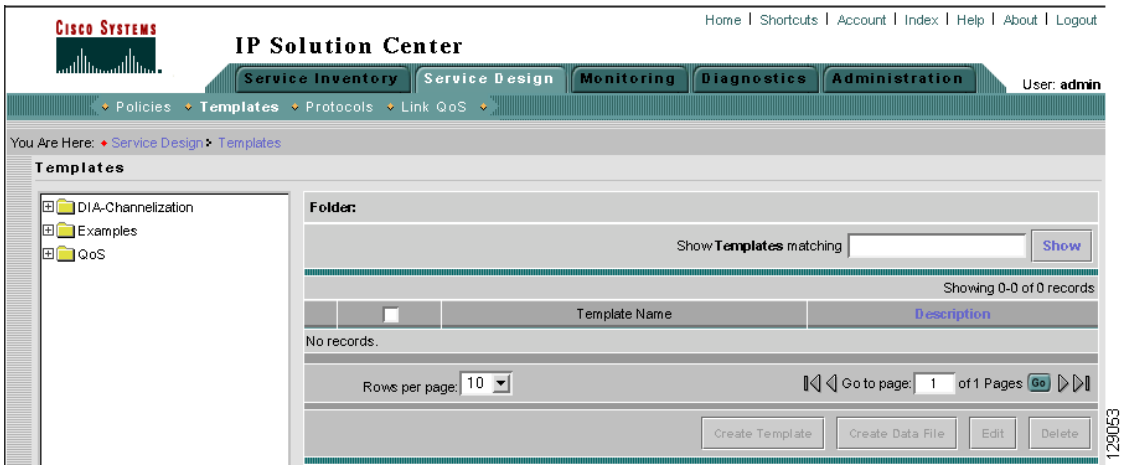
Template commands are treated independently from those associated with a service creation (Multi Protocol Label Switching (MPLS), Layer 2 Virtual Private Network (L2VPN), Virtual Private LAN Service (VPLS), Traffic Engineering (TE), and so on). Consequently, template commands must be removed separately from the device(s) during a service decommission. To remove prior template commands, a separate template is needed during a decommission process. Decommissioning a service request does not automatically remove the original template commands. A separate negate template needs to be added to the decommission process and the original templates must be removed. The negate template must contain the necessary NO commands to successfully remove any unwanted IOS commands added by the original template.

Note For additional information about template usage, see the Appendix D, "Template Usage".

To use Templates, follow these steps:

Step 1 Choose **Service Design > Templates** and you receive a window as shown in Figure 6-2, "Templates."

Figure 6-2 Templates



Template examples are shown in the left column. A complete list of template examples is specified in the "Template Examples" section on page 6-20. A complete list of Repository variables is shown in the "Summary of Repository Variables" section on page 6-22. An explanation of a tool for importing and exporting templates into and from an ISC database is given in the "Importing and Exporting Templates" section on page 6-34.

Step 2 Then you can do any of the following:

- View Templates Tree and Data Pane, page 6-3
- Create Folders and Subfolders, page 6-4
- Create Template, page 6-5
- Create Data File, page 6-13
- Edit, page 6-18
- Delete, page 6-19

View Templates Tree and Data Pane

When you choose **Service Design > Templates**, you receive a window as shown in Figure 6-3, "Tree and Data Pane Structure."

The Templates tree is in the left column. You can continue clicking the **+** sign next to each created folder and subfolder until you get to the last level of information. The last possible level is the template name. Data file information is not kept in the tree.

The right section of the window is the data pane. The name of the folder or template is in the upper-left corner. When you check the check box next to the template or data file information, the **Create Template**, **Create Data File**, **Edit**, or **Delete** buttons are enabled as described in the following sections.

When there are many templates in a folder or many data files in a template, the **Show Template Matching** or **Show Data File Matching** filter in the upper right-hand corner of the data pane can be very useful. For example, you might just want to work with templates or data files that start with **abc**. In this case, enter **abc*** in the field and then click the **Show** button. Only the templates or data files that start with **abc** appear.

You can also **View** configurations when the table displays data files.

Figure 6-3 Tree and Data Pane Structure

Create Folders and Subfolders

To create a new folder or subfolder, follow these steps:

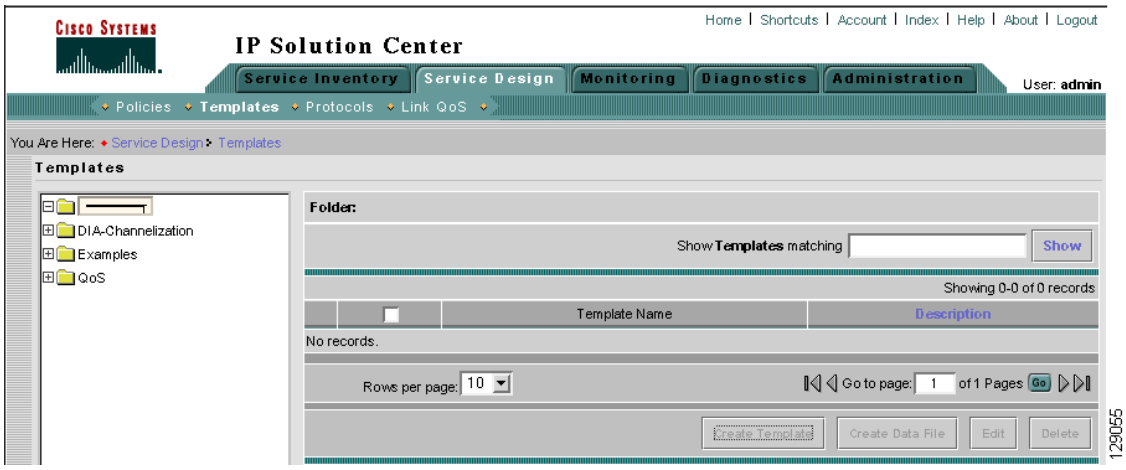
Step 1 Choose **Service Design** > **Templates**.

Step 2 In the **Templates** tree, right-click in the white area and choose **New** > **Folder** to create a new folder or right-click on an existing folder or subfolder and choose **New** > **Folder** to create a subfolder.

Note There is no limit to the number of levels of folders and subfolders you can create.

Step 3 In the new text field that appears in the **Templates** tree, type the new folder or subfolder name, as shown in the first entry of the **Templates** tree in Figure 6-4, "Folder Naming."

Figure 6-4 Folder Naming



Copying Folders or Subfolders

To copy a folder or subfolder and paste it into another folder or subfolder, follow these steps:

Step 1 Select a folder or subfolder and then right-click and you receive the opportunity to copy. Click **Copy**.

Step 2 Right-click on the folder or subfolder into which you want to paste the copied folder or subfolder and all its content and click **Paste**.

Step 3 You will see the new folder or subfolder and all its content in the selected location. You can edit and rename from there.

Create Template

You can either create a new template in an existing folder or you can create a new folder first and then create the template. To create a new folder, see the section "Create Folders and Subfolders".

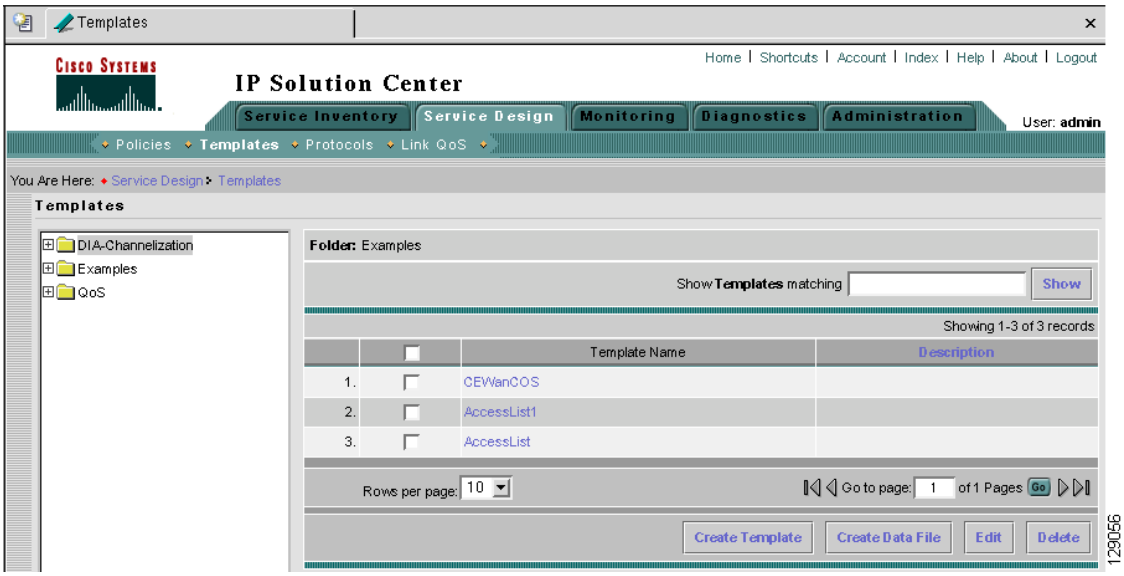
To create a new template, follow these steps:

Step 1 Choose **Service Design** > **Templates**.

Step 2 In the **Templates** tree, click on the folder in which you want to create a new template.

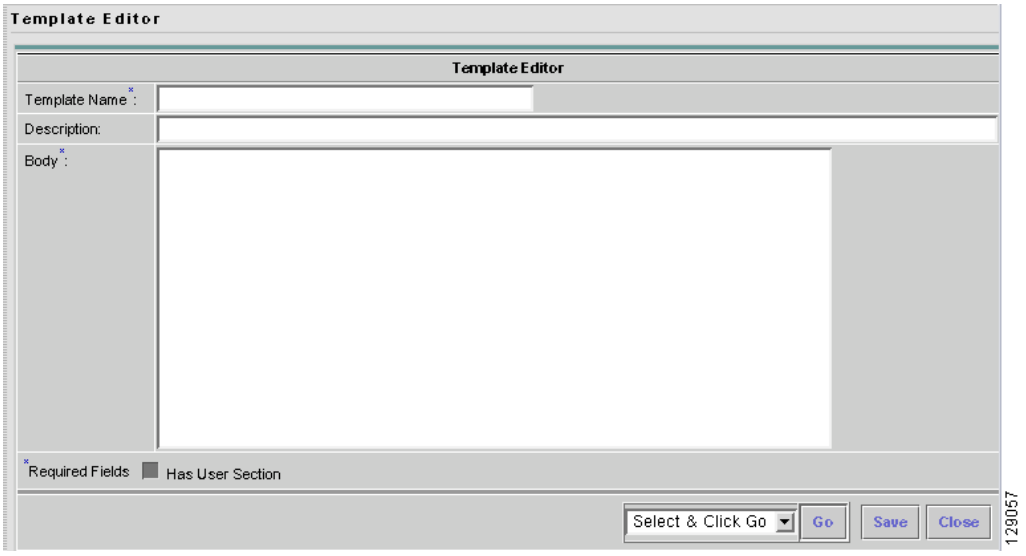
Step 3 A window appears as shown in Figure 6-5, "Folder with Existing Templates."

Figure 6-5 Folder with Existing Templates



Step 4 Click the **Create Template** button and you receive a window as shown in Figure 6-6, "Template Editor."

Figure 6-6 Template Editor



Step 5 Enter the following:

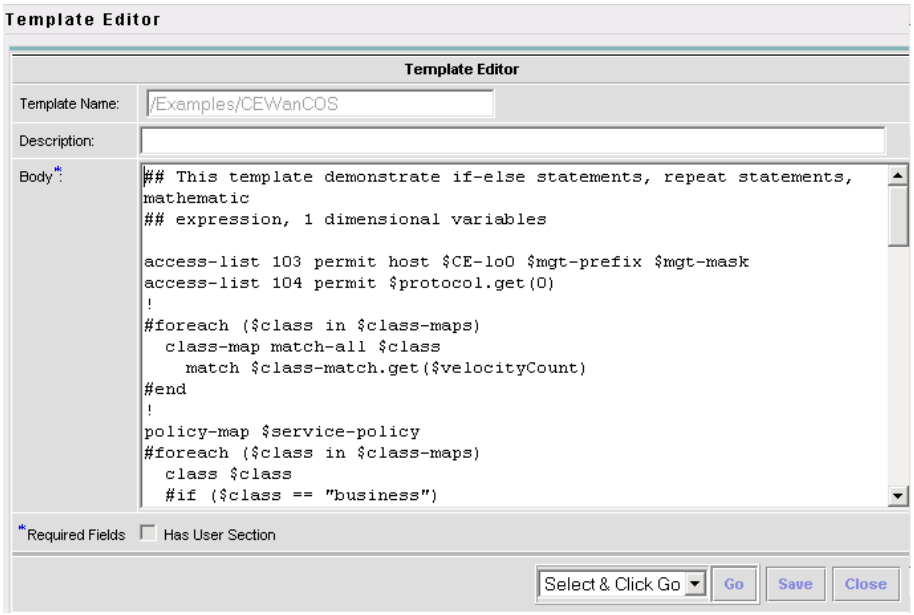
- **Template Name** (required) This must be a unique name within a folder. This name must begin with an alphabetic character and can only contain alphanumeric characters, underscores, and hyphens.
- **Description** (optional) You can enter any description here.
- **Body** (required) Enter the configuration text, Velocity Template Language (VTL) directives, and variables that you want included.

Note The VTL is explained at http://velocity.apache.org. For more specific information, you might like to navigate to http://velocity.apache.org/engine/devel/user-guide.html or http://velocity.apache.org/engine/devel/vtl-reference-guide.html.

Note For additional information about template usage, see the Appendix D, "Template Usage".

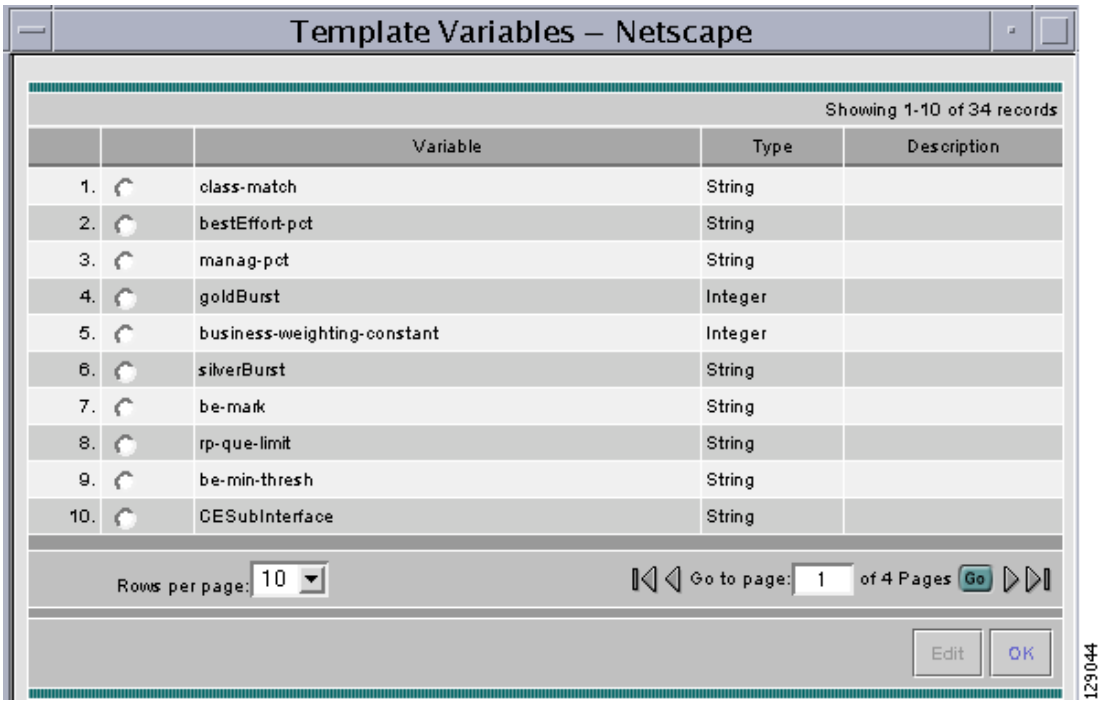
An example template is shown in Figure 6-7, "Example Template."

Figure 6-7 Example Template



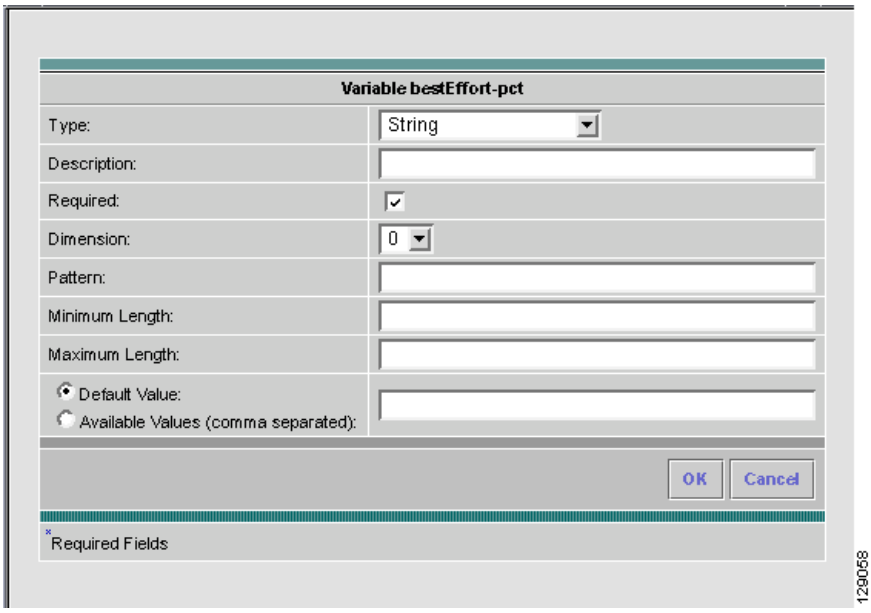
Step 6 Click the **Select & Click Go** drop-down list. If you want to validate the information you entered in Step 5, select **Validate** and then click the **Go** button. Otherwise, select **Variables** and then click the **Go** button and you receive a window as in Figure 6-8, Template Variables".

Figure 6-8 Template Variables



Step 7 Click the radio button for the Variable you want to edit and click **Edit**. You receive a window as shown in Figure 6-9, "Variable Definition—Type String."

Figure 6-9 Variable Definition—Type String



Step 8 In Figure 6-9, click the drop-down list for **Type** to receive the following choices:

- **String** Proceed to Step 9.
- **Integer** Proceed to Step 10.

- **Float** Proceed to Step 11.
- **IPv4 Address** Proceed to Step 12.
- **Sub-Template** Proceed to Step 13.

Step 9 The default Type to appear is **String**, a combination of ASCII characters considered as a group. The resulting Variable window is shown in Figure 6-9 and its attributes are as follows:

- **Description** (optional) You can enter any descriptive statement about this variable here.
- **Required** Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Dimension** Choose **0** (default), which indicates a scalar or enum variable; choose **1**, in which case the variable becomes a one-dimensional array; or choose **2**, in which case the variable becomes a two-dimensional array.
- **Pattern** (optional) Specify a regular expression pattern of the string. For example, a pattern of **isc[0-9]+** defines a string that starts with **isc** followed by one or more digits from **0** to **9**.
- **Minimum Length** (optional) If you specify a minimum length, the string cannot be less than the length specified here.
- **Maximum Length** (optional) If you specify a maximum length, the string cannot exceed the length specified here.
- Radio Button: **Default** (optional) If there is a default value for the specified variable, specify it here.
- Radio Button: **Available Values** (optional) Enter string values for this variable. Separate the values by commas.

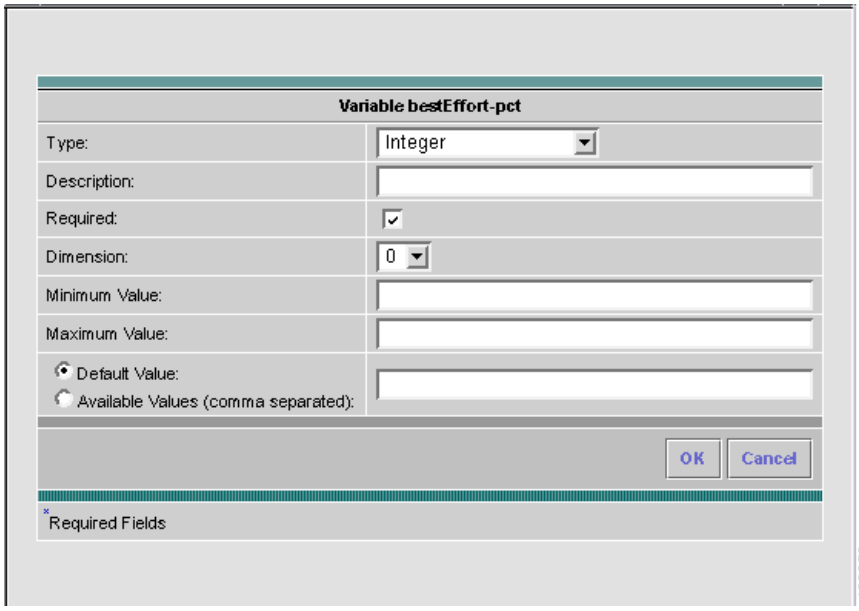
After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK** in a window such as Figure 6-8, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 6-6. Create a Data File is shown in the "Create Data File" section on page 6-13, **Edit** is shown in the "Edit" section on page 6-18, and **Delete** is shown in the "Delete" section on page 6-19.

Step 10 When you choose the Type **Integer**, a whole number, the resulting Variable window is shown in Figure 6-10 and its attributes are as follows:

- **Description** (optional) You can enter any descriptive statement about this variable here.
- **Required** Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Dimension** Choose **0** (default), which indicates a scalar or enum variable; choose **1**, in which case the variable becomes a one-dimensional array; or choose **2**, in which case the variable becomes a two-dimensional array.
- **Minimum Value** (optional) If you specify a minimum value, the integer cannot be less than the value specified here.
- **Maximum Value** (optional) If you specify a maximum value, the integer cannot exceed the value specified here.
- Radio Button: **Default** (optional) If there is a default value for the specified variable, specify it here.
- Radio Button: **Available Values** (optional) Enter string values for this variable. Separate the values by commas.

After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK** in a window such as Figure 6-8, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 6-6. Create a Data File is shown in the "Create Data File" section on page 6-13, **Edit** is shown in the "Edit" section on page 6-18, and **Delete** is shown in the "Delete" section on page 6-19.

Figure 6-10 Variable Definition—Type Integer

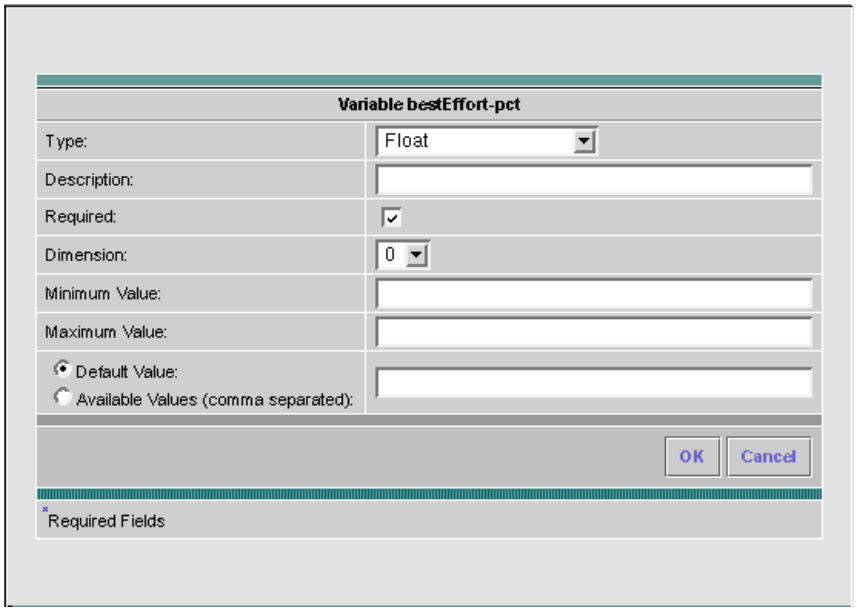


Step 11 When you choose the Type **Float**, a number that has no fixed number of digits before or after the decimal point, the resulting Variable window is shown in Figure 6-11 and its attributes are as follows:

- **Description** (optional) You can enter any descriptive statement about this variable here.
- **Required** Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Dimension** Choose **0** (default), which indicates a scalar or enum variable; choose **1**, in which case the variable becomes a one-dimensional array; or choose **2**, in which case the variable becomes a two-dimensional array.
- **Minimum Value** (optional) If you specify a minimum value, the floating point value cannot be less than the value specified here.
- **Maximum Value** (optional) If you specify a maximum value, the floating point value cannot exceed the value specified here.
- Radio Button: **Default** (optional) If there is a default value for the specified variable, specify it here.
- Radio Button: **Available Values** (optional) Enter string values for this variable. Separate the values by commas.

After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK** in a window such as Figure 6-8, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 6-6. Create a Data File is shown in the "Create Data File" section on page 6-13, **Edit** is shown in the "Edit" section on page 6-18, and **Delete** is shown in the "Delete" section on page 6-19.

Figure 6-11 Variable Definition—Type Float

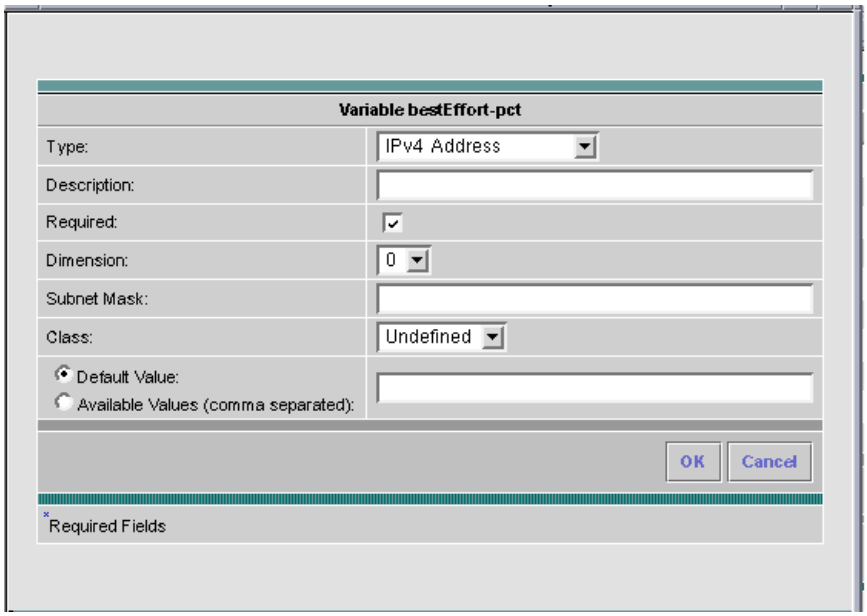


Step 12 When you choose the Type **IPv4 Address**, the resulting Variable window is shown in Figure 6-12 and its attributes are as follows:

- **Description** (optional) You can enter any descriptive statement about this variable here.
- **Required** Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Dimension** Choose **0** (default), which indicates a scalar or enum variable; choose **1**, in which case the variable becomes a one-dimensional array; or choose **2**, in which case the variable becomes a two-dimensional array.
- **Subnet Mask** (optional) Enter a valid subnet mask.
- **Class** (optional) Enter the class of the IP address. The options are: **Undefined**, **A**, **B**, or **C**.
- Radio Button: **Default** (optional) If there is a default value for the specified variable, specify it here.
- Radio Button: **Available Values** (optional) Enter string values for this variable. Separate the values by commas.

After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK** in a window such as Figure 6-8, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 6-6. Create a Data File is shown in the "Create Data File" section on page 6-13, **Edit** is shown in the "Edit" section on page 6-18, and **Delete** is shown in the "Delete" section on page 6-19.

Figure 6-12 Variable Definition—Type IPv4



Step 13 When you choose the Type **Sub-Template**, you instantiate one subtemplate into the Main template. The resulting Variable window is shown in Figure 6-13 and its attributes are as follows:

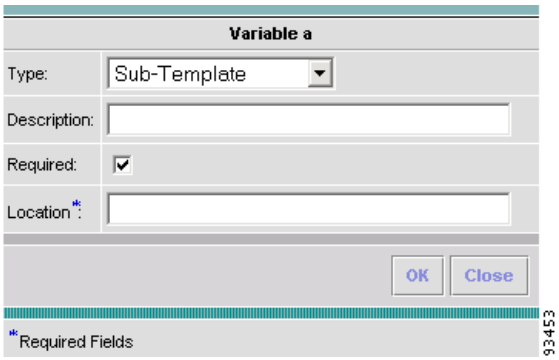
- **Description** (optional) You can enter any descriptive statement about this variable here.
- **Required** Leave the default of the checked check box if this variable is required. Otherwise, uncheck it.
- **Location** (required) Enter the full path name of the parent template. For example **/test2/testyy**.

The variable varName is defined as the subtemplate type (by selecting **Variables** and clicking **Go**). The Sub-Template defined earlier is called and you must provide the subtemplate path. The syntax is as follows:

$<*varName*>**.callWithDatafile(**<*DatafileName*>**)**

After you enter all the data, click **OK** to accept this information for the specified variable; continue editing all variables you want to change in this same way, then click **OK** in a window such as Figure 6-8, which now includes these updated variables; click **Save** and then **Close** or click **Close** and when asked, agree to **Save** for a window such as Figure 6-6. Create a Data File is shown in the "Create Data File" section on page 6-13, **Edit** is shown in the "Edit" section on page 6-18, and **Delete** is shown in the "Delete" section on page 6-19.

Figure 6-13 Variable Definition—Type Sub-Template



Copying Templates

To copy a template and paste it into another folder, follow these steps:

Step 1 Select a template and then right-click and you receive the opportunity to copy. Click **Copy**.

Step 2 Right-click on the folder into which you want to paste the copied template and all its data files and click **Paste**.

Step 3 You will see the new template and all its data files in the selected location. You can edit and rename from there.

Create Data File

You can create a new data file from an existing template. If the template you want is not available, go to the "Create Template" section on page 6-5.

To create a data file, follow these steps:

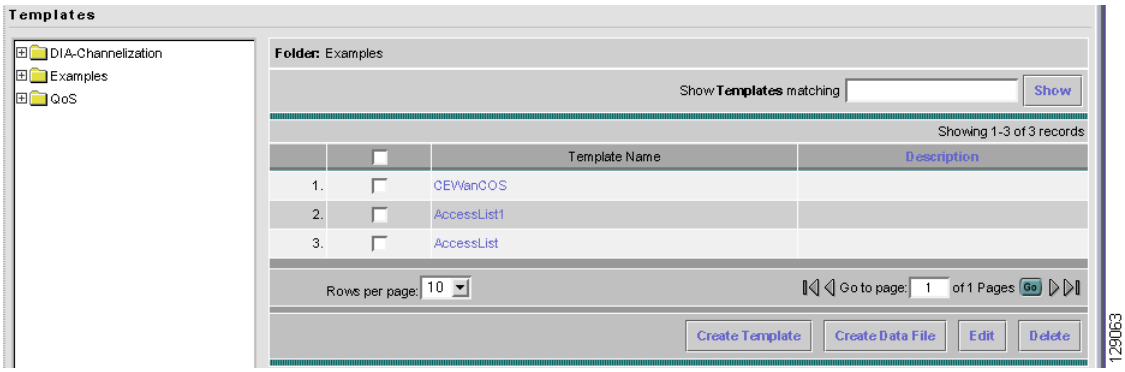
Step 1 Choose **Service Design** > **Templates**.

Step 2 In the **Templates** tree in the left part of your window, do one of the following

1. Left-click on the folder or subfolder in which the template for which you want to create a data file exists or
2. Click on the **+** next to the folder of choice and then click on the template for which you want to create a data file.

Step 3 If you chose 1. in Step 2, a window appears as shown in Figure 6-14, "Choose Existing Template > Create Data File."

Figure 6-14 Choose Existing Template > Create Data File

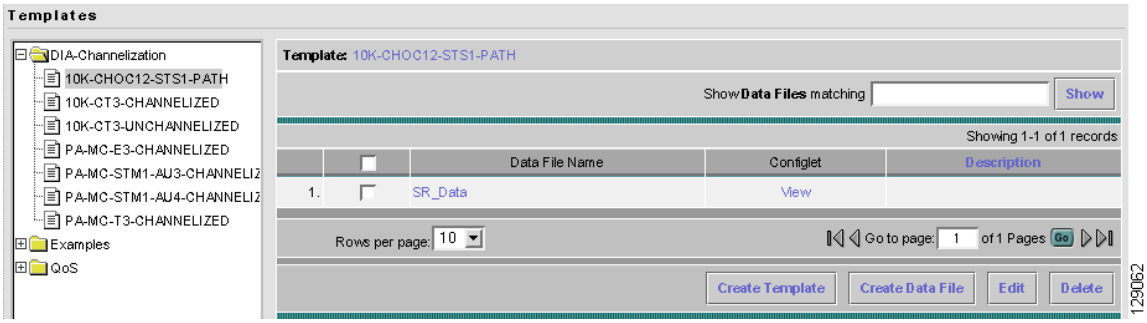


Check the check box for the template for which you want to create a data file and click **Create Data File**. Then proceed to Step 5.

Otherwise, proceed to Step 4.

Step 4 If you chose 2. in Step 2, the buttons appear as shown in Figure 6-15, "Choose Existing Template > Create Data File."

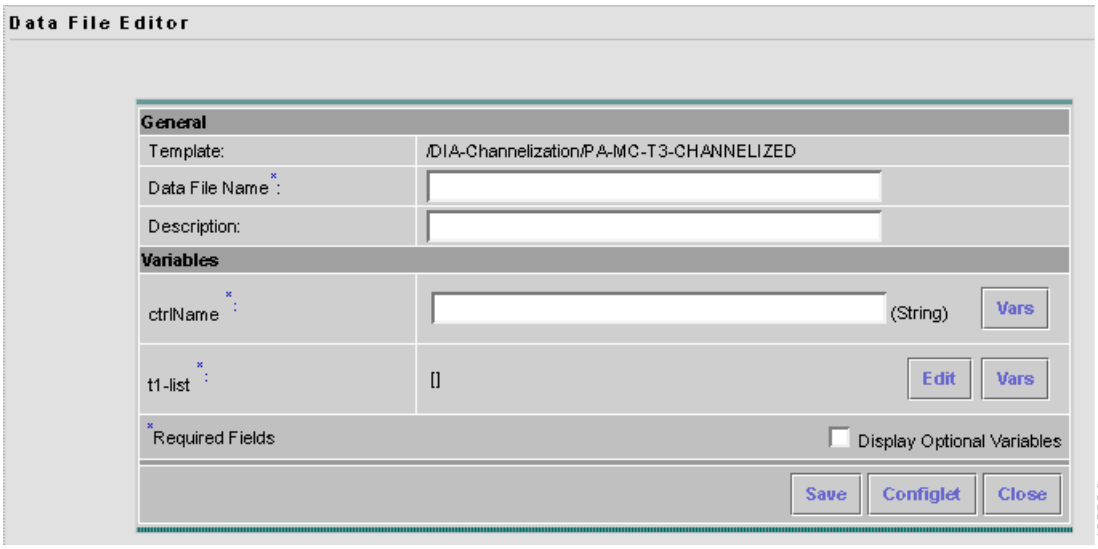
Figure 6-15 Choose Existing Template > Create Data File



Click **Create Data File** and proceed to Step 5.

Step 5 An example of a window that appears is shown in Figure 6-16, "Template Data File Editor."

Figure 6-16 Template Data File Editor



Step 6 In the **General** area, fill in the following:

- **Data File Name** (required) This must be a unique name. This name must begin with an alphabetic character and can only contain alphanumeric characters and the underscore.
- **Description** (optional) Enter any description that helps you identify this data file.

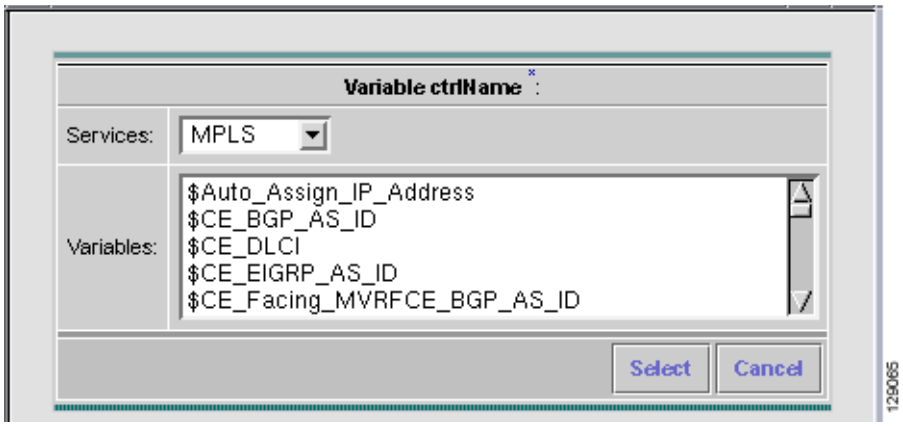
Step 7 In the example in Figure 6-16, in the **Variables** area, **cntrlName** is a string variable (**Dimension** defined when the template was created was **0**); you can also create a one-dimensional array (**Dimension** defined when the template was created was **1**); and **t1-list** is a two-dimensional array (**Dimension** defined when the template was created was **2**).

If **t1-list** is a Dynamic Java Class variable, you *must* enter the entire Java Class package name. For example: com.cisco.isc.class_name.

Note **cntrlName** can *only* be a string variable.

Step 8 If you click **Vars** as shown in Figure 6-16, you receive a window as shown in Figure 6-17, "Template Data File Editor."

Figure 6-17 Template Data File Editor



Click the **Services** drop-down list to have access to variables for:

- **MPLS**
- **L2VPN**
- **VPLS**

Then click the entry in **Variables** that you want to use and click **Select**.

If you have a **0** dimensional entry (set as **Dimension 0** when creating a template), you can only enter variables in the provided field.

Step 9 When you click **Edit**, as shown in Figure 6-16, the resulting window depends on whether you are editing a **1** or **2** dimensional array.

Proceed to Step 10 for information about a **1** dimensional array.

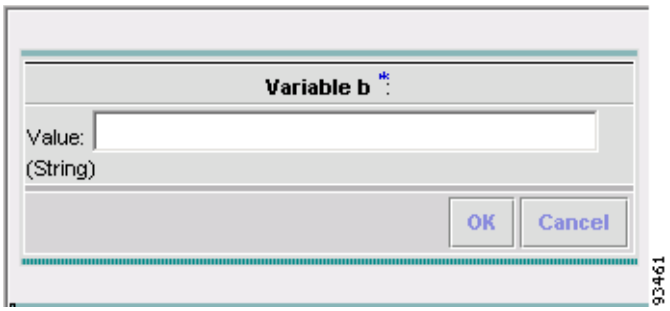
Proceed to Step 13 for information about a **2** dimensional array.

Step 10 For a one-dimensional array (set as **Dimension 1** when creating the template), when you click **Edit**, you receive a window as shown in Figure 6-18, "Editing a One-Dimensional Array."

Figure 6-18 Editing a One-Dimensional Array

Step 11 To add a variable, click **Add** and a window, as shown in Figure 6-19, "Adding a Variable," appears in which you can add the variable. Then click **OK**.

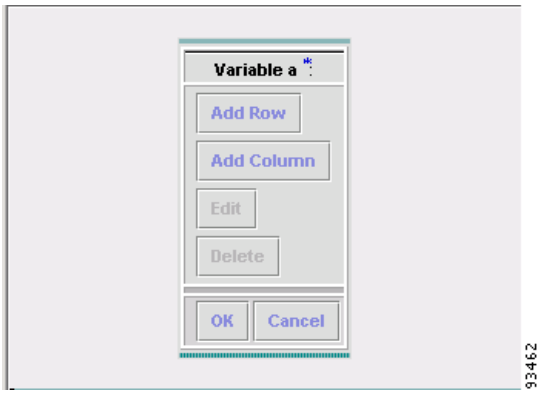
Figure 6-19 Adding a Variable



Step 12 To edit or delete a variable, highlight the variable in Figure 6-18 and click **Edit** or **Delete**. For **Edit** you receive a window as shown in Figure 6-19. Then click **OK**. For **Delete**, *be sure* you want to delete. After you click **Delete**, it automatically occurs and the window is updated. Proceed to Step 19.

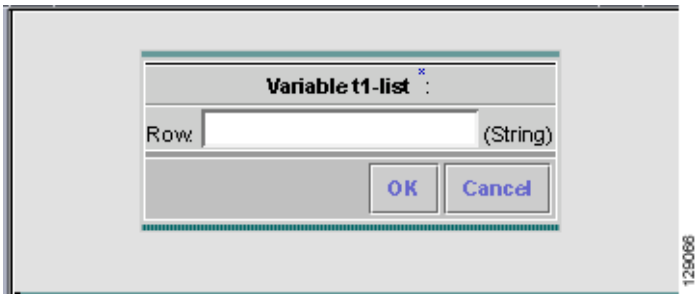
Step 13 For a two-dimensional array (set as **Dimension 2** when creating the template), when you click **Edit**, you receive a window as shown in Figure 6-20, "Editing a Two-Dimensional Array.

Figure 6-20 Editing a Two-Dimensional Array



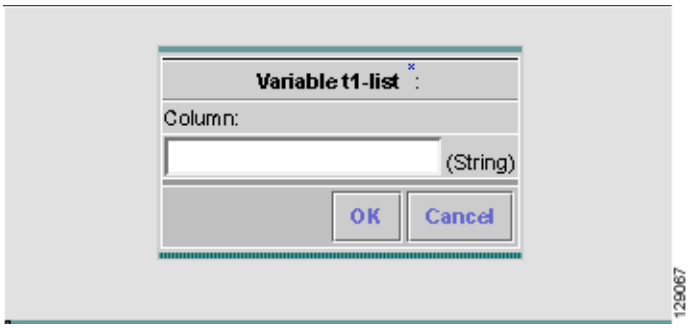
Step 14 Click **Add Row** in Figure 6-20 and a window, as shown in Figure 6-21, "Enter Row Information," appears. Enter a value and click **OK**.

Figure 6-21 Enter Row Information



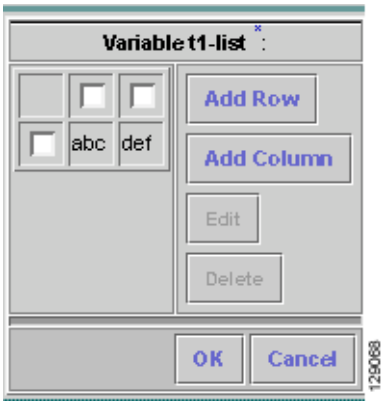
Step 15 Click **Add Column** in Figure 6-20 and a window as shown in Figure 6-22, "Enter Column Information," appears. Enter a value and click **OK**.

Figure 6-22 Enter Column Information



Step 16 A resulting window, as shown in Figure 6-23, "Two-Dimensional Array Results," appears.

Figure 6-23 Two-Dimensional Array Results



Step 17 You can check any of the check boxes (toggles) and you can then **Edit** or **Delete** that row or column. You can also continue to **Add Row** and **Add Column** as shown in Step 15 and Step 16, respectively.

Step 18 When you complete setting up your two-dimensional array, click **OK** in Figure 6-23.

Step 19 A window as shown in Figure 6-16 is updated to reflect the new data file information.

Step 20 You can then click **Save** and then **Close** to save this information and close this file; click **Configure** to show the configuration file; or click **Close** and then be sure to click **OK**, if you want to save the information you have created. If you do not want to save this information, click **Close** and then click **Cancel**.

Edit

To edit a Template or Data File, follow these steps:

Step 1 Choose **Service Design** > **Templates**.

Step 2 In the **Templates** tree, left-click on the folder or subfolder in which the template you want to edit exists or the template in which the data file you want to edit exists. Alternatively, when the name in the upper left corner of the data pane is a template, you can click on the template name to edit the template.

Step 3 To edit a template, a window appears as shown in Figure 6-24, "Choose Existing Template > Edit." To edit a data file, a window appears as shown in Figure 6-25, "Choose Existing Data File > Edit."

Figure 6-24 Choose Existing Template > Edit

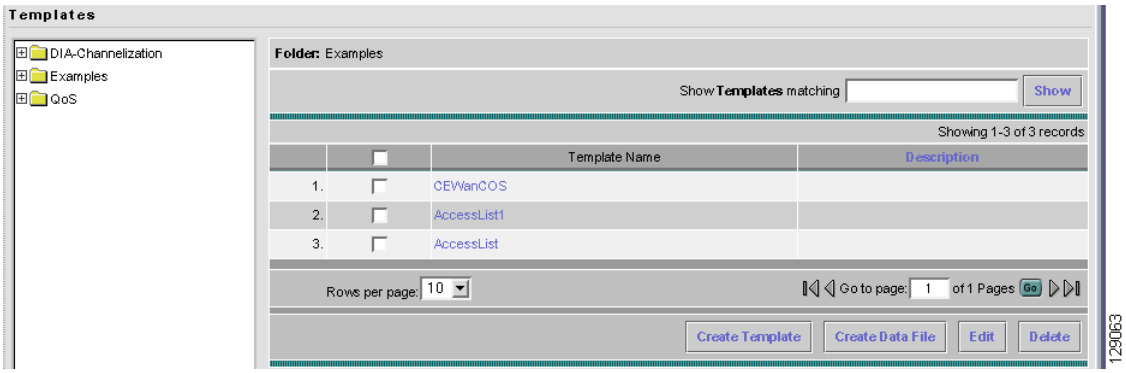
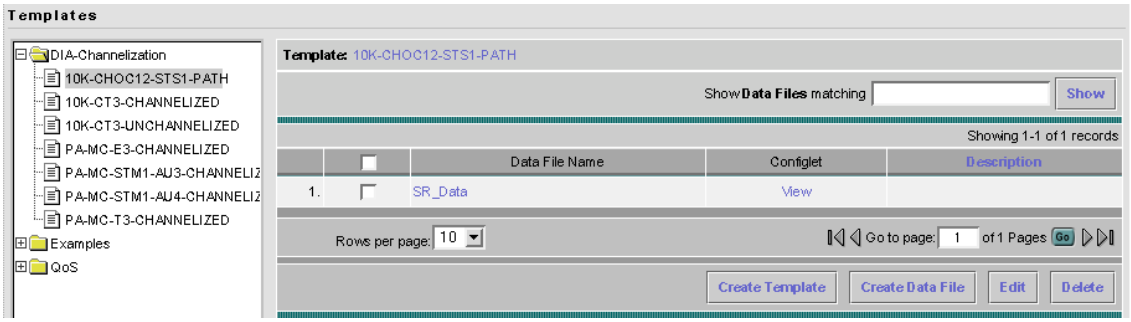


Figure 6-25 Choose Existing Data File > Edit



Step 4 Check the check box for the template or data file you want to edit.

Note For a data file, there is a **Configlet** column in which you can click **View** to view the configuration file.

Step 5 Click **Edit**.

Step 6 When editing a template, you receive a window as shown in Figure 6-6, "Template Editor." Then proceed as in Step 5Step 6 in the Create Template section. When editing a data file, you receive a window as shown in Figure 6-15, "Choose Existing Template > Create Data File." Then proceed as in Step 5 in the Create Data File section.

Delete

To delete a Template or Data File, follow these steps:

Step 1 Choose **Service Design > Templates**.

Step 2 In the **Templates** tree, left-click on the folder or subfolder in which the template you want to delete exists or the template in which the data file you want to delete exists.

Step 3 To delete a template, a window appears as shown in Figure 6-26, "Choose Existing Template > Delete." To delete a data file, a window appears as shown in Figure 6-27, "Choose Existing Data File > Delete."

Figure 6-26 Choose Existing Template > Delete

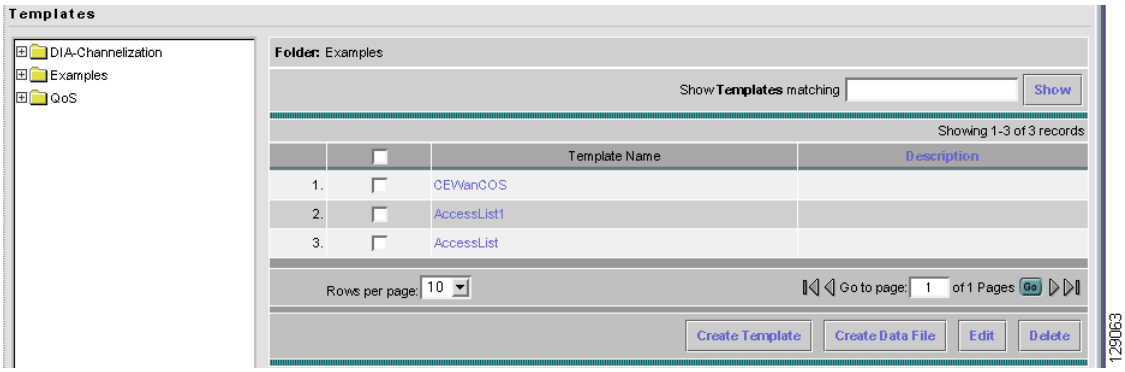
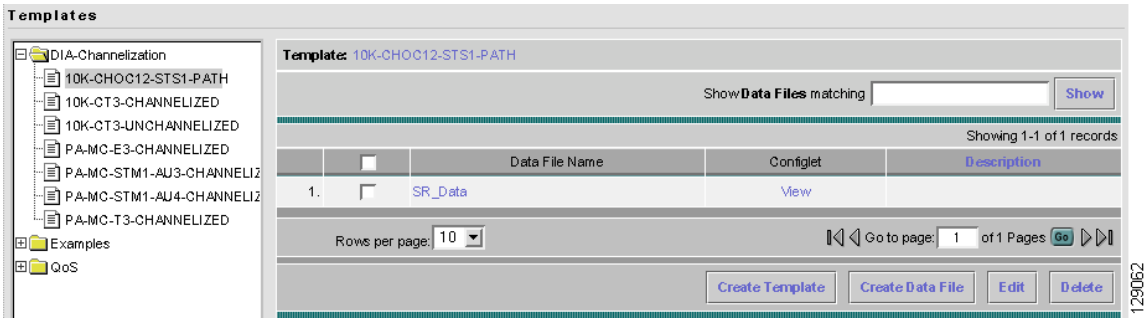


Figure 6-27 Choose Existing Data File > Delete



Step 4 Check the check box for the template or data file you want to delete.

Note For a data file, there is a **Configlet** column in which you can click **View** to view the configuration file.

Step 5 Click **Delete**.

Step 6 You receive an updated window as shown in Figure 6-26, "Choose Existing Template > Delete" or Figure 6-27, "Choose Existing Data File > Delete" with the deleted template or data file no longer available.

Template Examples

In the left column, the hierarchy pane, of **Service Design > Templates**, as shown in Figure 6-2, "Templates," template examples appear. See Table 6-1, "Template Examples and Their Descriptions."

Table 6-1 Template Examples and Their Descriptions

Folder	Template	Description
DIA-Channelization	10K-CHOC12-STS1-PATH	Sample template to break down channelized OC12 to STS-1 paths.
	10K-CT3-CHANNELIZED	Sample template creates T1 out of channnelized T3 line card.
	10K-CT3-UNCHANNELIZED	Sample template Creates either a fullrate T3 or a subrate T3 interface out of a channelized T3.
	PA-MC-E3-CHANNELIZED	Sample template Creates E1 (channel groups) out of E3.
	PA-MC-STM1-AU3-CHANNELIZE	Sample template Creates E1 (channel groups) out of TUG-2. This template uses AU-3 AUG mapping that further creates TUG-2s.
	PA-MC-STM1-AU4-CHANNELIZE	Sample template Creates E1 (channel groups) out of TUG-2. This template uses AU-4 AUG mapping that creates TUG-3s and TUG-2s.
	PA-MC-T3-CHANNELIZED	Sample template Creates T1 (channel groups) out of T3.
Examples	AccessList	Demonstrates templates with nested repeat loop and multi-dimension variable.
	AccessList1	Demonstrates the simplest template variable substitution.
	CEWanCOS	Demonstrates if-else statements, repeat statements, mathematical expressions, and one-dimensional variables.
QoS/L2/ATM	CLP_Egress	Sample template to demonstrate the setting of qos_group and ATM Cell Loss Priority at the output of an interface.
	CLP_Ingress	Sample template sets MPLS experimental bit of the ATM Cell marked with Cell Loss Priority, at the input of an interface.
QoS/L2/Ethernet	3400_Egress	
QoS/L2/FrameRelay	classification	Sample template to demonstrate the bandwidth reservation based on FrameRelay DLCI value.

Summary of Repository Variables

This section contains the following tables:

- Table 6-2 on page 6-22, "L2VPN Repository Variables"
- Table 6-3 on page 6-25, "MPLS Repository Variables"
- Table 6-4 on page 6-32, "VPLS Repository Variables"

Table 6-2 provides a summary of the L2VPN Repository variables available from ISC Templates.

Table 6-2 L2VPN Repository Variables

Repository Variable	Dimension	Description
AC_Loopback_Address	0	PE loopback address also known as the router ID
CE_DLCI	0	DLCI value on CE for Frame Relay encapsulation
CE_Encap	0	Encapsulation of the CE interface
CE_Intf_Desc	0	Interface description for the CE interface
CE_Intf_Main_Name	0	Major interface name for the CE interface
CE_Intf_Shutdown	0	Shutdown flag for the CE interface
CE_VCD	0	VCD value on CE for ATM encapsulation
CE_VCI	0	VCI value on CE for ATM encapsulation
CE_Vlan_ID	0	VLAN ID on CE for Ethernet encapsulation
CE_VPI	0	VPI value on CE for ATM encapsulation
L2VPNCLECeFacingEncapsulation	0	Encapsulation of the UNI
L2VPNCLECeFacingInterfaceName	0	Name of the UNI
L2VPNCLEPeFacingEncapsulation	0	Encapsulation of the NNI (should always be dot1q)
L2VPNCLEPeFacingInterfaceName	1	Name of the NNI (uplinks) (the number can be more than 1 in case of a ring topology, hence any array)
L2VPNDFBIT_SET	0	Indicates not to fragment the bit set (for L2TPv3 only)
L2VPNDynamicModeUseDefaults	0	Dynamic session setup using ISC default values (for L2TPv3 only)
L2VPN_intf_main_name	1	The main interface name for a CE or PE port
L2VPNIP_PMTU	0	Enable the discovery of the path MTU for tunneled traffic (for L2TPv3 only)
L2VPNIP_TOS	0	Configure the value of the TOS byte in IP headers of tunneled packets or reflects the TOS byte value from the inner IP header (for L2TPv3 only)
L2VPNIP_TTL	0	Configure the value of the time to live byte in the IP headers (for L2TPv3 only)

Table 6-2 L2VPN Repository Variables (continued)

Repository Variable	Dimension	Description
L2VPNL2TP_CLASS_NAME	0	The L2TP class name to overwrite the default L2TP class name (for L2TPv3 only)
L2VPNL2TPv3Sequence	0	Specifies the direction in which sequencing of data packets in a pseudo wire is enabled (for L2TPv3 only)
L2VPNLocalCookieHighValue	0	Specifies the last 4 bytes of the value that the peer PE must include in the cookie field of incoming L2TP packets (for L2TPv3 only)
L2VPNLocalCookieLowValue	0	Specifies the first 4 bytes of the value that the peer PE must include in the cookie field of incoming L2TP packets (for L2TPv3 only)
L2VPNLocalCookieSize	0	Specifies the size (0, 4, or 8) of the cookie field of incoming L2TP packets (for L2TPv3 only)
L2VPNLocalLoopBack	1	The head of the L2TPv3 tunnel
L2VPNLocalSessionId	0	Specifies the ID for the local L2TPv3 session (for L2TPv3 only)
L2VPNLocalSwitchLoopBack1	1	The loopback1 for the local switch (for L2TPv3 only)
L2VPNLocalSwitchLoopBack2	1	The loopback2 for the local switch (for L2TPv3 only)
L2VPNRemoteCookieHighValue	1	Specifies the last 4 bytes of the value that this PE must include in the cookie field of incoming L2RP packets (for L2TPv3 only)
L2VPNRemoteCookieLowValue	1	Specifies the first 4 bytes of the value that this PE must include in the cookie field of incoming L2RP packets (for L2TPv3 only)
L2VPNRemoteCookieSize	1	Specifies the size (0, 4, or 8) of the cookie field of outgoing L2TP packets (for L2TPv3 only)
L2VPNRemoteLoopback	0	The tail of the L2TPv3 tunnel
L2VPNRemoteSessionID	1	Specifies the ID for the remote L2TPv3 session (for L2TPv3 only)
L2VPNSessionSetupMode	0	Defines how the L2TPv3 session is set up (static or dynamic) (for L2TPv3 only)
L2VPNTransportMode	0	Defines how the L2TPv3 data is transferred (for Frame Relay: DLCI or Port; for ATM: VP or VC) (for L2TPv3 only)
L2VPNUniMajorInerfaceName	0	The main interface name of the UNI
L2VPNVcId	0	The virtual circuit ID of the L2TPv3 or AToM tunnel
PE_DLCI	0	DLCI value on PE for Frame Relay encapsulation

Table 6-2 L2VPN Repository Variables (continued)

Repository Variable	Dimension	Description
PE_Encap	0	Encapsulation of the PE interface
PE_Intf_Desc	0	Interface description for the PE interface
PE_Intf_Main_Name	0	Major interface name for the PE interface
PE_VCD	0	VCD value on PE for ATM encapsulation
PE_VCI	0	VCI value on PE for ATM encapsulation
PE_Vlan_ID	0	VLAN ID on PE for Ethernet encapsulation
PE_VPI	0	VPI value on PE for ATM encapsulation
PseudoWire_Class_Type_Of_Core	0	Core type of the Service Provider over which L2VPN is provisioned
Uni_Aging	0	Length of time the MAC address can stay on the port security table
Uni_Cdp_Enable	0	Flag to enable or disable layer 2 tunnelling on a Cisco Discover Protocol (CDP)
Uni_Cdp_Threshold	0	Number of packets per second to be received before the interface is shut down for the CDP protocol
Uni_Mac_Address	0	Number of MAC addresses allowed for port security
Uni_Port_Security	0	Flag to enable or disable security on a UNI interface
Uni_Protocol_Tunnelling	0	Flag to enable or disable Layer 2 Bridge Protocol Data Unit (BPDU) protocol tunnelling on a UNI interface
Uni_Recovery_Interval	0	Amount of time to wait before recovering a UNI port
Uni_Shutdown	0	Flag indicating whether the User Network Interface (UNI) is shutdown
Uni_Speed	0	Value of the UNI link speed
Uni_Stp_Enable	0	Flag to enable or disable layer 2 tunnelling on a Spanning Tree Protocol (STP)
Uni_Stp_Threshold	0	Flag to enable or disable layer 2 tunnelling on an STP
Uni_Violation_Access	0	Action taken when a port security violation is detected
Uni_Vtp_Enable	0	Flag to enable or disable layer 2 tunnelling on a VLAN Trunk Protocol (VTP)
Uni_Vtp_Threshold	0	Flag to enable or disable layer 2 tunnelling on a VTP

Table 6-3 provides a summary of the MPLS Repository variables available from ISC Templates.

Table 6-3 MPLS Repository Variables

Repository Variable	Dimension	Description
Advertised_Routes_To_CE	2	List of one or more IP addresses of the advertised static route to be placed on the PE to define the CE's address space
CE_BGP_AS_ID	0	BGP AS ID on a CE when the routing protocol between a CE and a PE is BGP
CE_DLCI	0	DLCI value on CE for Frame Relay encapsulation
CE_EIGRP_AS_ID	0	EIGRP AS ID on a CE when the routing protocol between a CE and a PE is EIGRP
CE_Facing_MVRFCE_BGP_AS_ID	0	BGP AS ID on an MVRFCE when the routing protocol between a CE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_DLCI	0	DLCI value on CE facing MVRFCE interface for Frame Relay encapsulation, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_EIGRP_AS_ID	0	EIGRP AS ID on an MVRFCE when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_Intf	0	Name of the CE facing interface on an MVRFCE, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_Intf_Address	0	IP address assigned to the CE facing MVRFCE interface, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_Intf_Encap	0	Encapsulation for CE facing of an MVRFCE interface, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_Intf_Name	0	Name of the CE facing MVRFCE interface, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_Intf_Type	0	Interface type for CE facing of an MVRFCE interface, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_Ospf_Process_ID	0	OSPF process ID on MVRFCE when the routing protocol between a CE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_Tunnel_Src_ Addr	0	Tunnel source address on CE facing MVRFCE interface for GRE encapsulation when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_VCD	0	VCD value on CE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE

Table 6-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
CE_Facing_MVRFCE_VCI	0	VCI value on CE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_VLAN_ID	0	VLAN ID on CE facing MVRFCE interface for Ethernet encapsulation, when an MPLS link includes an MVRFCE
CE_Facing_MVRFCE_VPI	0	VPI value on CE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE
CE_Intf_Address	0	IP address assigned to the CE interface
CE_Intf_Encap	0	Encapsulation of the CE interface
CE_Intf_Name	0	Name of the CE interface
CE_MVRFCE_Bandwidth_Metric_For_ Redistribution	0	Bandwidth metric for redistribution of EIGRP when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFC
CE_MVRFCE_BGP_AS_ID	0	BGP AS ID on a CE when the routing protocol between a CE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE
CE_MVRFCE_Delay_Metric_For_ Redistribution	0	Delay metric for redistribution of EIGRP when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFC
CE_MVRFCE_EIGRP_AS_ID	0	EIGRP AS ID on a CE when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE
CE_MVRFCE_Loading_Metric_For_ Redistribution	0	Loading metric for redistribution of EIGRP when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFC
CE_MVRFCE_MTU_Metric_For_ Redistribution	0	MTU metric for redistribution of EIGRP when the routing protocol between a CE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFC
CE_MVRFCE_Ospf_Process_ID	0	OSPF process ID on CE when the routing protocol between a CE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE
CE_Ospf_Process_ID	0	OSPF process ID on CE when the routing protocol between a CE and a PE is OSPF
CE_Tunnel_Src_Addr	0	Tunnel source address on CE for GRE encapsulation

Table 6-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
CE_VCD	0	VCD value on CE for ATM encapsulation
CE_VCI	0	VCI value on CE for ATM encapsulation
CE_Vlan_ID	0	VLAN ID on CE for Ethernet encapsulation
CE_VPI	0	VPI value on CE for ATM encapsulation
Export_Map	0	Name of the export map associated with the VRF
Extra_CE_Loopback_Required	0	Flag to indicate whether an extra loopback request is required on the CE
Import_Map	0	Name of the import map associated with the VRF
Is_Default_Info_Originate	0	Flag to indicate whether the **default-information originate** command for BGP on the PE when STATIC is a running protocol between a CE and a PE
Is_Default_Routes_Sent_To_CE	0	Flag to indicate whether the default routes are sent to a remote CE
Join_Grey_Mgmt_Vpn	0	Flag to indicate whether MPLS will join a Grey Management VPN
Max_route_threshold	0	Percentage of the maximum number of routes that can be imported into the VRF
Max_Routes	0	Maximum number of routes than can be imported into the VRF
MPLSExportRouteTargets	1	List of Route Targets that are exported for a particular VRF associated with the MPLS VPN link
MPLSImportRouteTargets	1	List of Route Targets that are imported for a particular VRF associated with the MPLS VPN link
MPLSCLEPeFacingInterfaceName	0	The name of the interface on the device facing the PE for that particular MPLS VPN link
MPLSCLEPeFacingEncapsulation	0	The encapsulation of the interface on the device facing the PE for that particular MPLS VPN link
MPLSCLECeFacingInterfaceName	0	The name of the interface on the device facing the CE for that particular MPLS VPN link
MPLSCLECeFacingEncapsulation	0	The encapsulation of the interface on the device facing the CE for that particular MPLS VPN link
MPLSCeInterfaceMask	0	The mask of the IP address assigned to the CE interface for a particular MPLS VPN link
MPLSPeInterfaceMask	0	The mask of the IP address assigned to the PE interface for a particular MPLS VPN link

Table 6-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
MPLSCeLoopbackAddress	0	The IP address of the extra CE loopback address for a particular MPLS VPN link
MVRFCE_CE_Advertised_Routes_To_CE	2	List of one or more IP addresses of the advertised static route to be placed on the PE to define the CE's address space, when the MPLS link includes an MVRFCE
MVRFCE_CE_IP_Unnumbered	0	Flag to indicate whether the MVRCE to CE link is unnumbered, when an MPLS link includes an MVRFCE
MVRFCE_CE_Is_Default_routes_Sent_To_CE	0	Flag to indicate whether the default routes are sent to a remote CE, when an MPLS link includes an MVRFCE
MVRFCE_CE_NBR_ALLOW_AS_IN	0	AllowASIn flag when the routing protocol between a CE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE
MVRFCE_CE_NBR_AS_OVERRIDE	0	ASOverride flag when the routing protocol between a CE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE
MVRFCE_CE_Ospf_Area_Number	0	OSPF area number when the routing protocol between a CE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE
MVRFCE_CE_Routes_To_Reach_Other_Sites	2	List of one or more IP addresses to specify the static routes to put on the CE, when the MPLS link includes an MVRFCE
MVRFCE_CE_Routing_Protocol	0	Routing protocol between MVRFCE and CE
PE_BGP_AS_ID	0	BGP AS ID on a PE when the routing protocol between a CE and a PE is BGP
PE_Cable_Both_Helper_Address_List	1	List of DHCP server IP addresses to which both cable modem and host UDP broadcasts are forwarded
PE_Cable_Modem_Helper_Address_list	1	List of DHCP server IP addresses to which cable modem UDP broadcasts are forwarded
PE_Cable_Modem_Host_Helper_Address_List	1	List of DHCP server IP addresses to which host UDP broadcasts are forwarded
PE_Cable_Modem_Secondary_Address_List	1	List of cable modem secondary addresses for cable interfaces
PE_CE_Bandwidth_Metric_For_Redistribution	0	Bandwidth metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP
PE_CE_Delay_Metric_For_Redistribution	0	Delay metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP

Table 6-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
PE_CE_IP_Unnumbered	0	Flag to indicate whether the PE to CE link is unnumbered
PE_CE_Loading_Metric_For_ Redistribution	0	Loading metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP
PE_CE_MTU_Metric_For_ Redistribution	0	MTU metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP
PE_CE_NBR_Allow_AS_In	0	AllowASIn flag when the routing protocol between a CE and a PE is BGP
PE_CE_NBR_AS_Override	0	ASOverride flag when the routing protocol between a CE and a PE is BGP
PE_CE_Ospf_Area_Number	0	OSPF area number when the routing protocol between a CE and a PE is OSPF
PE_CE_Reliability_Metric_For_ Redistribution	0	Reliability metric for redistribution of EIGRP when the routing protocol between a CE and a PE is EIGRP
PE_CE_Routing_Protocol	0	Routing protocol between PE and CE
PE_DLCI	0	DLCI value on PE for Frame Relay encapsulation
PE_EIGRP_AS_ID	0	EIGRP AS ID on a PE when the routing protocol between a CE and a PE is EIGRP
PE_Facing_MVRFCE_BGP_AS_ID	0	BGP AS ID on an MVRFCE when the routing protocol between a PE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_DLCI	0	DLCI value on PE facing MVRFCE interface for Frame Relay encapsulation, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_EIGRP_AS_ID	0	EIGRP AS ID on an MVRFCE when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_Intf	0	Name of the PE facing interface on an MVRFCE, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_Intf_Address	0	IP address assigned to the PE facing MVRFCE interface, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_Intf_Encap	0	Encapsulation for PE facing of an MVRFCE interface, when an MPLS link includes an MVRFCE

Table 6-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
PE_Facing_MVRFCE_Intf_Name	0	Name of the PE facing MVRFCE interface, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_Intf_Type	0	Interface type for PE facing of an MVRFCE interface, when an MPLS link includes an MVRFCE
PE_FACING_MVRFCE_OSPF_ Process_ID	0	OSPF process ID on an MVRFCE when the routing protocol between a PE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_Tunnel_Src_Addr	0	Tunnel source address on PE facing MVRFCE interface for GRE encapsulation when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_VCD	0	VCD value on PE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_VCI	0	VCI value on PE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_VLAN_ID	0	VLAN ID on PE facing MVRFCE interface for Ethernet encapsulation, when an MPLS link includes an MVRFCE
PE_Facing_MVRFCE_VPI	0	VPI value on PE facing MVRFCE interface for ATM encapsulation, when an MPLS link includes an MVRFCE
PE_Intf_Address	0	IP address assigned to the PE interface
PE_Intf_Desc	0	Interface description for the PE interface
PE_Intf_Encap	0	Encapsulation of the PE interface
PE_Intf_Name	0	Name of the PE interface
PE_Intf_Shutdown	0	Shutdown flag for the PE interface
PE_IS_Cable_Modem_Maintenance_ Interface	0	Flag to indicate whether the interface is a maintenance interface
PE_MVRFCE_Bandwidth_Metric_For_ Redistribution	0	Bandwidth metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE
PE_MVRFCE_BGP_AS_ID	0	BGP AS ID on a PE when the routing protocol between a PE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE
PE_MVRFCE_Delay_Metric_For_ Redistribution	0	Delay metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE

Table 6-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
PE_MVRFCE_EIGRP_AS_ID	0	EIGRP AS ID on a PE when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE
PE_MVRFCE_IP_Unnumbered	1	Flag to indicate whether the PE to MVRFCE link is unnumbered, when an MPLS link includes an MVRFCE
PE_MVRFCE_Loading_Metric_For_ Redistribution	0	Loading metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE
PE_MVRFCE_MTU_Metric_for_ redistribution	0	MTU metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE
PE_MVRFCE_NBR_ALLOW_AS_IN	0	AllowASIn flag when the routing protocol between a PE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE
PE_MVRFCE_NBR_AS_OVERRIDE	0	ASOverride flag when the routing protocol between a PE and an MVRFCE is BGP, when an MPLS link includes an MVRFCE
PE_MVRFCE_Ospf_Area_Number	0	OSPF area number when the routing protocol between a PE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE
PE_MVRFCE_OSPF_Process_ID	0	OSPF process ID on PE when the routing protocol between a PE and an MVRCE is OSPF, when an MPLS link includes an MVRFCE
PE_MVRFCE_Reliability_Metric_For_ Redistribution	0	Reliability metric for redistribution of EIGRP when the routing protocol between a PE and an MVRFCE is EIGRP, when an MPLS link includes an MVRFCE
PE_MVRFCE_Routing_Protocol	0	Routing protocol between PE and MVRFCE, when an MPLS link includes an MVRFCE
PE_OSPF_PROCESS_ID	0	OSPF process ID on PE when the routing protocol between a CE and a PE is OSPF
PE_Tunnel_Src_Addr	0	Tunnel source address on PE for GRE encapsulation
PE_VCD	0	VCD value on PE for ATM encapsulation
PE_VCI	0	VCI value on PE for ATM encapsulation
PE_Vlan_ID	0	VLAN ID on PE for Ethernet encapsulation
PE_VPI	0	VPI value on PE for ATM encapsulation
rd	0	Route Distinguisher value for the VRF

Table 6-3 MPLS Repository Variables (continued)

Repository Variable	Dimension	Description
Redistribute_Connected	0	Flag to indicate whether the connected routes are redistributed into BGP on the PE
Redistribute_Static	0	Flag to indicate whether the static routes are redistributed into BGP on the PE
Redistributed_Protocol	1	List of routing protocols to be redistributed
Rip_Metrics	0	Metric for redistribution associated with RIP
Routes_To_Reach_Other_Sites	2	List of one or more IP addresses to specify the static routes to put on the CE.
vrfName	0	Name of the VRF

Table 6-4 provides a summary of the VPLS Repository variables available from ISC Templates.

Table 6-4 VPLS Repository Variables

Repository Variables	Dimension	Description
VPLSCeEncapsulation	0	The encapsulation of the CE interface for a particular VPLS link
VPLSCeInterfaceName	0	The name of the CE interface for a particular VPLS link
VPLSCeMajorInterfaceName	0	The name of a major interface on a CE for a particular VPLS link
VPLSCLECeFacingEncapsulation	0	The encapsulation of interfaces for a particular device facing the CE
VPLSCLECeFacingInterfaceName	0	The interface name for a particular device facing the CE (the number can be more than 1 in case of a ring topology, hence any array)
VPLSCLEPeFacingEncapsulation	0	The encapsulation of interfaces for a particular device facing the PE
VPLSCLEPeFacingInterfaceName	1	The list of interface names for a particular device facing the PE (the number can be more than 1 in case of a ring topology, hence any array)
VPLSDisableCDP	0	The flag to specify if the CDP has been disabled on a UNI for a particular VPLS link
VPLSFilterBPDU	0	The flag to specify whether the BPDUs will be filtered on a UNI for a particular VPLS link
VPLSPeEncapsulation	0	The encapsulation of the PE interface for a particular VPLS link
VPLSPeInterfaceDescription	0	The description assigned to the PE interface for a particular VPLS link
VPLSPeInterfaceName	0	The name of the PE interface for a particular VPLS link

Table 6-4 VPLS Repository Variables (continued)

Repository Variables	Dimension	Description
VPLSPeMajorInterfaceName	0	The name of a major interface on a PE for a particular VPLS link
VPLSPeNeighbors	1	The list of PE POPs participating in a particular VPLS VPN
VPLSPeVfiName	0	The VFI name assigned to a particular VPLS instance existing on the PE POP
VPLSPeVlanId	0	The VLAN ID assigned to the PE for a particular VPLS link
VPLSPeVpnId	0	The VPN ID assigned to a particular VPLS VPN
VPLSSystemMTU	0	The maximum MTU value for a packet arriving on a UNI for a particular VPLS link
VPLSTunnelCDPEnable	0	The flag to specify if the CDP packets will be tunneled to the remote site for a particular VPLS link
VPLSTunnelCDPThreshold	0	The threshold value assigned for a CDP protocol before a violation action is reported on a UNI for a particular VPLS link
VPLSTunnelRecoveryInterval	0	Interval for the UNI to recover from a shutdown scenario
VPLSTunnelSTPEnable	0	The flag to specify if the STP packets will be tunneled to the remote site for a particular VPLS link
VPLSTunnelSTPThreshold	0	The threshold value assigned for a STP protocol before a violation action is reported on a UNI for a particular VPLS link
VPLSTunnelVTPEnable	0	The flag to specify if the VTP packets will be tunneled to the remote site for a particular VPLS link
VPLSTunnelVTPThreshold	0	The threshold value assigned for a VTP protocol before a violation action is reported on a UNI for a particular VPLS link
VPLSUniAging	0	The aging timer set on a UNI for a particular VPLS link
VPLSUniDuplex	0	The duplex assigned to the UNI for a particular VPLS link
VPLSUniMajorInterfaceName	0	The name of a major interface on a UNI device for a particular VPLS link
VPLSUniMaxMacAddress	0	The maximum number of Mac addresses that can be learned on a UNI for a particular VPLS link
VPLSUniPortSecurity	0	The port security option on a UNI for a particular VPLS link

Table 6-4 VPLS Repository Variables (continued)

Repository Variables	Dimension	Description
VPLSUniProtocolTunneling	0	The flag to specify if the protocols will be tunneled to the remote site for a particular VPLS link
VPLSUniSecureMacAddresses	1	The explicit list of Mac addresses that can be learned on a UNI for a particular VPLS link
VPLSUniShutdown	0	The shutdown flag on a UNI for a particular VPLS link
VPLSUniSpeed	0	The speed assigned to the UNI for a particular VPLS link
VPLSUniViolationAction	0	The violation action option on a UNI for a particular VPLS link
VPLSUseNativeVlan	0	The flag to specify if the native VLAN will be used on a UNI for a particular VPLS link

Importing and Exporting Templates

The **importExportTemplateDB** tool is available to import and export templates into and from an ISC database. You can import or export the complete or partial template database by specifying appropriate arguments. You can find this tool at: **$ISC_HOME/bin/importExportTemplateDB.sh**.

Enter the following:

importExportTemplateDB.sh <*admin_user_id*> <*password*> [<*other_arguments*>]

where:

<*admin_user_id*> is user identifier for someone with the **admin** role.

<*password*> is the password for the one with the **admin** role.

<*other_arguments*> is any combination of the following arguments separated by a space:

-nooverwrite
If you choose to use this **nooverwrite** argument, to prevent the overwriting of existing templates in the database, it must precede all other arguments and must be in the third position after <*admin_user_id*> and <*password*>.

Note The default (when **nooverwrite** is not specified) is to overwrite the templates.

-exp_db <*dest-dir*>
Use this argument to export all templates and datafiles in the database, where <*dest-dir*> is the destination directory to which you want to export.

-imp_db <*src-dir*>
Use this argument to import all the files in <*src-dir*> into the database, where <*src-dir*> is the source directory from which you want to import. The files in <*src-dir*> are created by the **exp_db** process.

-exp_template_folder <*src-folder-path*> <*dest-dir*>
Use this argument to export a database template folder and its subfolders, where **<***src-folder-path*> is the full path of the template folder to export and <*dest-dir*> is the directory where to place the exported files.

-imp_template_folder <*src-dir***> <***dest-folder***>**
Use this argument to import all files in **<***src-dir***>** into the database, where **<***src-dir***>** is the source directory to import, and <*dest-folder*> is the destination import template folder.

-imp_template <*srcfile***> <***dest-folder***> <***template-name***>**
Use this argument to import a template into the database, where **<***srcfile***>** is the full path of the template to import, <*dest-folder*> is the full path of the parent folder, and <*template-name*> is the template name in the database.

-imp_datafile <*srcfile***> <***dest-template***> <***datafile-name***>**
Use this argument to import a template datafile into the database, where **<***srcfile***>** is the full path of the datafile to import, <*dest-template*> is the full path of the parent template, and <*datafile-name*> is the datafile name in the database.

-exp_template <*template-pathname***> <***output-file***>**
Use this argument to export the database template to a file, where **<***template-pathname***>** is the full path of the template to export, and <*output-file*> is the output filename.

-exp_datafile <*datafile-pathname***> <***output-file***>**
Use this argument to export a template datafile to a file, where **<***datafile-pathname***>** is the full path of the template datafile to export, and <*output-file*> is the output filename.